Target-Adaptive Neural Architecture Search in YOLOv9 Machine Vision Models

Seok Bin Son[†], Yeryeong Cho[†], Joongheon Kim[†], and Soohyun Park[§]
[†]Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
[§]Division of Computer Science, Sookmyung Women's University, Seoul, Republic of Korea
E-mails: {lydiasb,joyena0909,joongheon}@korea.ac.kr, soohyun.park@sookmyung.ac.kr

Abstract—**Recent advances in real-time object detection have been driven by YOLO models, which effectively balance accuracy and speed. However, architectures optimized under a fixed search configuration often show limited adaptability when applied to diverse deployment targets. To address this limitation, this paper introduces a target-adaptive YOLOv9 neural architecture search (NAS) algorithm that applies NAS to the neck block of YOLOv9 through a once-for-all supernet. By defining multiple search scopes with varying exploration ranges, the framework enables the automatic generation of sub-networks tailored to different target requirements without additional retraining. Experimental results demonstrate that the small configuration achieves 73.4% precision and 53.2% mAP50, confirming an effective balance between accuracy and model efficiency. The proposed approach facilitates scalable deployment across diverse target settings by flexibly adjusting the architectural search range.**

Index Terms—**Neural Architecture Search, NAS, YOLO, YOLOv9NAS, Target-Adaptive YOLOv9NAS**

I. Introduction

Recent progress in computer vision has concentrated on structural innovations that enhance object detection performance [1], [2]. The "you only look once (YOLO)" family has established itself as a prominent framework due to its single-pass detection capabilities, demonstrating both speed and accuracy in real-time applications [3], [4]. While deeper or more complex architectures can improve accuracy, such designs often lack adaptability across different deployment targets with varying requirements [5]. Neural architecture search (NAS) provides an effective solution by automatically discovering optimal network architectures tailored to specific datasets and objective functions, thereby reducing reliance on manual design while improving efficiency and accuracy [6]–[9]. However, existing NAS methods typically employ a single fixed search configuration, which limits their ability to generate architectures suited to multiple target scenarios. As a result, models optimized under a single search setting may not generalize well when applied to diverse target configurations.

This paper presents a target-adaptive YOLOv9NAS algorithm that addresses this limitation by enabling architecture search under multiple exploration ranges. The algorithm applies NAS to the neck block of YOLOv9, which integrates multi-scale feature representations and plays a central role in determining detection accuracy [4]. The neck block provides manageable search-space complexity while preserving high exploration efficiency, making it a suitable region for architectural optimization. The proposed framework adopts a once-for-all supernet that trains a unified network containing all candidate sub-network paths, allowing the direct extraction of sub-networks corresponding to different target scopes without additional retraining [10]. Four search scopes (i.e., small, medium, large, and full) are defined by varying the exploration range within the supernet. Each module in the neck block corresponds to a feature fusion stage, where combinations of kernel sizes (i.e., 3×3, and 5×5) and activation functions (i.e., ReLU, LeakyReLU, and Mish) are explored. By adjusting the search range for each target scope, the framework automatically identifies architectures that reflect the characteristics of each target setting [2], [6].

The contributions of this paper are twofold. First, NAS is applied to the YOLOv9 neck block to jointly optimize structural efficiency and detection performance within a well-defined search space. Second, a target-adaptive NAS framework is introduced, enabling the generation of multiple target-specific architectures from a single once-for-all supernet by varying the exploration ranges across search scopes. This approach supports flexible deployment across diverse target scenarios while maintaining stable detection performance.

II. Related Work

A. Neural Architecture Search

NAS automates network design through three fundamental components: search space definition, search strategy, and performance evaluation. Traditional NAS methods train each candidate architecture independently, incurring prohibitive computational and temporal costs [8], [9], [11]. One-shot NAS addresses these inefficiencies by constructing a supernet that encompasses all candidate architectures, allowing for the simultaneous evaluation of multiple sub-networks through weight sharing within a unified training process. Differentiable architecture search (DARTS) employs continuous relaxation to the search space and optimizes architecture parameters via gradient descent. However, this approach is hindered by excessive

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-00561377) and also by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00766, Development of Integrated Development Framework that supports Automatic Neural Network Generation and Deployment optimized for Runtime Environment) *(Corresponding author: Joongheon Kim)*

memory consumption and training instability. ProxylessNAS employs binary path selection, where only one candidate path is activated per training iteration, resulting in superior memory efficiency and training stability. This methodology has influenced subsequent one-shot NAS frameworks that strike a balance between search effectiveness and computational tractability.

B. *YOLOv9 Algorithm*

YOLO constitutes a state-of-the-art framework for real-time object detection, performing simultaneous predictions of object locations and class probabilities through a single forward pass [3], [4]. The architecture consists of three components: the backbone extracts feature maps from input images, the neck integrates multi-scale features for enhanced contextual representation, and the head generates final predictions for object localization and classification. Successive YOLO versions have demonstrated progressive improvements in detection accuracy and computational efficiency. YOLOv9, the latest iteration, extends the YOLOv8 architecture with enhanced performance on high-resolution images and complex visual scenes [4]. NAS-guided optimization in YOLOv9 enables improved real-time inference while preserving the balance between detection accuracy and computational speed. This architectural refinement addresses the trade-off between model capacity and inference efficiency across diverse deployment scenarios.

III. Target-Adaptive YOLOv9NAS Algorithm

A. *YOLOv9NAS*

YOLOv9NAS applies NAS exclusively to the neck block of YOLOv9, automating structural selection within the multi-scale feature integration module to optimize the component most critical to the speed-accuracy trade-off [4]. The neck is selected as the search target due to its computational intensity and capacity for substantial performance gains without disrupting the overall detection pipeline. The search space comprises two design dimensions: convolution kernel sizes (i.e., 3×3, and 5×5) and activation functions (i.e., ReLU, LeakyReLU, and Mish). This deliberate restriction prevents exponential growth in training time and computational cost while maintaining sufficient architectural diversity based on validated reference structures. Model training adopts a one-shot supernet framework where a single supernet containing all candidate operations is trained once with shared weights across all paths. Binary path activation ensures that only one path is computed per iteration, thereby enhancing memory efficiency and training stability. The architecture parameters represent the selection probabilities for each path. The loss function incorporates hardware-aware regularization based on computational cost, feature map resolution, and memory usage, prioritizing constraint-aware optimization over unconstrained exploration. In conclusion, a lightweight model is extracted by retaining only the selected path from the trained supernet, while the unchanged backbone and head enable direct reuse of existing pipelines and pretrained weights. This modular design enables straightforward adaptation to diverse deployment environments through the replacement of neck subnetworks alone, preserving preprocessing and postprocessing chains. YOLOv9NAS is characterized by four key properties: localized neck optimization, compact search space with kernel and activation variations, one-shot training with binary path selection, and explicit hardware cost normalization for resource-constrained deployment.

B. *Target-Adaptive YOLOv9NAS Algorithm*

Conventional NAS methods typically optimize architectures under a single fixed search setting, which limits their adaptability when applied to diverse deployment targets with varying structural requirements. The proposed target-adaptive NAS framework addresses this limitation by utilizing a once-for-all supernet that can generate multiple sub-networks from a single trained model. Four target scopes (i.e., small, medium, large, and full) are defined by assigning different exploration ranges within the supernet. During training, one scope is sampled per mini-batch, and the architecture probability distribution is normalized over candidate paths belonging to the sampled scope's designated search range. After training, the path with the highest selection probability within each scope is fixed, resulting in four distinct architectures tailored to their corresponding target settings. This design enables scope-specific model generation while maintaining architectural diversity across different target configurations.

The training process consists of four sequential stages. Supernet pre-training stabilizes shared parameters across multiple epochs using the standard YOLOv9 loss. The subsequent target-adaptive search phase samples one scope per mini-batch and activates a single path corresponding to the exploration range defined for that scope through binary gating during backpropagation. Path fixation and fine-tuning then identify the highest-probability path for each scope and refine accuracy through additional training. Finally, the export procedure produces specialized weights for the small, medium, large, and full configurations, all sharing the same backbone and head while varying only the neck block. Although binary path selection may risk premature convergence by concentrating probabilities on a limited set of paths early in training, stabilization strategies such as entropy normalization, balanced scope sampling, and random path masking can effectively mitigate this issue. By isolating the neck block as the search region while preserving the backbone and head, the framework enables modular adaptation across multiple target scenarios without altering the overall detection pipeline.

IV. Experiments

This section describes the experimental setup and results for the target-adaptive YOLOv9NAS algorithm. YOLOv9-M serves as the baseline architecture, with NAS exploration confined to module-level operations within the neck block. All experiments employ consistent datasets, training configurations, and hyperparameters to ensure a fair comparison across four target scopes (i.e., small, medium, large, and full), corresponding to distinct hardware constraints.

TABLE I: Target-Adaptive YOLOv9NAS Optimal Architecture

Scope	Architecture
Small	"m_13": 1 , "m_16": 0, "m_19": 0, "m_22": 0, "m_28": 0, "m_31": 0, "m_34": 2, "m_37": 1
Medium	"m_13": 2 , "m_16": 0, "m_19": 2, "m_22": 0, "m_28": 0, "m_31": 2, "m_34": 2, "m_37": 0
Large	"m_13": 2 , "m_16": 3, "m_19": 3, "m_22": 0, "m_28": 0, "m_31": 3, "m_34": 0, "m_37": 1
Full	"m_13": 2 , "m_16": 0, "m_19": 2, "m_22": 0, "m_28": 0, "m_31": 0, "m_34": 2, "m_37": 0

TABLE II: The Performance of Target-Adaptve YOLOv9NAS

Scope	Precision (%)	Recall (%)	mAP50 (%)
Small	73.4	46.6	53.2
Medium	68.9	41.8	47.6
Large	68.5	41.3	46.9
Full	67.1	42.8	48.5

A. Experimental Setup

Experiments were conducted on an NVIDIA RTX 4090 GPU with an input resolution of 640 × 640 and a batch size of 16. The Adam optimizer was configured with a learning rate of 0.01, momentum of 0.937, and weight decay of 0.0005, with all scopes trained and evaluated on the COCO dataset.

B. Performance

Table I presents the optimal architecture configurations identified through NAS exploration for each target scope. The small scope primarily selects simpler operations, indicating that its restricted exploration range naturally leads to more compact structural choices. In contrast, the large and full scopes, which allow broader exploration, tend to adopt more diverse and structurally expressive operations. This pattern demonstrates that the search process reflects the characteristics of each scope by adjusting the exploration range, rather than relying on a single, unified configuration. Through this mechanism, the target-adaptive NAS produces distinct architectures for each scope from a shared Supernet, ensuring structural diversity across multiple target settings.

Table II summarizes detection performance using precision, recall, and mAP50 metrics across the four target scopes. The small scope achieves the highest performance, with 73.4% precision and 53.2% mAP50, resulting from a simplified architecture that reduces redundant paths and enhances feature utilization. The medium, large, and full scopes show slight variations in precision while maintaining stable recall, reflecting structural differences derived from their respective exploration ranges. The full scope achieves 48.5% mAP50 and demonstrates consistent performance due to its broader architectural configuration. Overall, compact architectures tend to yield higher precision within restricted exploration settings, while larger scopes provide more extensive feature combinations enabled by broader search space definitions.

V. Conclusion

This paper presents a target-adaptive YOLOv9NAS algorithm that applies NAS to the neck block by defining different exploration ranges for multiple target scopes. The once-for-all supernet enables the automatic generation of four distinct architectures without additional retraining, allowing each scope to reflect structural characteristics derived from its designated search range. Experimental results show that the small scope achieves 73.4% precision and 53.2% mAP50, demonstrating the effectiveness of a compact architecture produced through restricted exploration. These findings confirm that adjusting the search range for different target scopes is a practical and scalable approach for generating diverse object detection models within a unified NAS framework.

References

[1] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the International Conference on Computer Vision, (ICCV)*, Montreal, QC, Canada, October 2021, pp. 9992–10 002.

[2] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext V2: Co-designing and scaling convnets with masked autoencoders," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, Vancouver, BC, Canada, June 2023, pp. 16 133–16 142.

[3] C. Wang, A. Bochkovskiy, and H. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, Vancouver, BC, Canada, June 2023, pp. 7464–7475.

[4] C. Wang, I. Yeh, and H. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *Proc. of the European conference on computer vision, (ECCV)*, vol. 15089, Milan, Italy, September 2024, pp. 1–21.

[5] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proc. of the International Conference on Machine Learning, (ICML)*, vol. 139, Virtual Event, July 2021, pp. 10 096–10 106.

[6] M. Chen, H. Peng, J. Fu, and H. Ling, "Autoformer: Searching transformers for visual recognition," in *Proc. of the IEEE/CVF International Conference on Computer Vision, (ICCV)*, Montreal, QC, Canada, October 2021, pp. 12 250–12 260.

[7] C. Gong, D. Wang, M. Li, X. Chen, Z. Yan, Y. Tian, Q. Liu, and V. Chandra, "Nasvit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training," in *Proc. of the International Conference on Learning Representations, (ICLR)*, Virtual Event, April 2022.

[8] S. B. Son and S. Park, "Q-RLONAS: Towards efficient quantum neural architecture search," in *Proc. IEEE International Conference on Quantum Computing and Engineering (QCE)*, Albuquerque, New Mexico, USA, September 2025.

[9] S. B. Son, S. Y.-C. Chen, J. Kim, and S. Park, "Filtered one-shot training for quantum architecture search," in *Proc. ACM Conference on Information and Knowledge Management (CIKM)*, Seoul, South Korea, November 2025.

[10] X. Dai, A. Wan, P. Zhang, B. Wu, Z. He, Z. Wei, K. Chen, Y. Tian, M. Yu, P. Vajda, and J. E. Gonzalez, “Fbnetv3: Joint architecture-recipe search using predictor pretraining,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, Virtual, June 2021, pp. 16 276–16 285.

[11] S. Park, S. B. Son, Y. K. Lee, S. Jung, and J. Kim, “Two-stage architectural fine-tuning for neural architecture search in efficient transfer learning,” *Electronics Letters*, vol. 59, no. 24, p. e13066, December 2023.